UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/718,297	11/20/2003	Timothy Gerrit Deboer	CA920020055US1	9784

35525 7590 04/28/2010

IBM CORP (YA)
C/O YEE & ASSOCIATES PC
P.O. BOX 802333
DALLAS, TX 75380

EXAMINER	
SIKRI, ANISH	

ART UNIT	PAPER NUMBER
2443	

NOTIFICATION DATE	DELIVERY MODE
04/28/2010	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

ptonotifs@yeeiplaw.com

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/718,297
Filing Date: November 20, 2003
Appellant(s): DEBOER ET AL.

Cathrine K Kinslow
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 1/20/10 appealing from the Office action mailed 8/21/09.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

No amendment after final has been filed.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The examiner has no comment on the appellant's statement of the grounds of rejection to be reviewed on appeal. Every ground of rejection set forth in the Office action from which the appeal is taken (as modified by any advisory actions) is being maintained by the examiner except for the grounds of rejection (if any) listed under the subheading

"WITHDRAWN REJECTIONS." New grounds of rejection (if any) are provided under the subheading "NEW GROUNDS OF REJECTION."

NEW GROUND(S) OF REJECTION

A new grounds of rejection under 35 U.S.C. 101 has been added for claims 28-31 and 35.

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claim(s) 28, 29-31, 35 are rejected under 35 USC 101 since the claims are directed to non-statutory subject matter. Claim(s) 29-31 recite computer readable media which appear to cover both transitory and non-transitory embodiments. The United States Patent and Trademark Office (USPTO) is required to give claims their broadest reasonable interpretation consistent with the specification during proceedings before the USPTO. *See In re Zletz*, 893 F.2d 319 (Fed. Cir. 1989) (during patent examination the pending claims must be interpreted as broadly as their terms reasonably allow). The broadest reasonable interpretation of a claim drawn to a computer readable medium (also called machine readable medium and other such variations) typically covers forms of non-transitory tangible media **and** transitory propagating signals *per se* in view of the ordinary and customary meaning of computer readable media, particularly when the specification is silent. *See* MPEP 2111.01. When the broadest reasonable

interpretation of a claim covers a signal *per se*, the claim **must** be rejected under 35 U.S.C. § 101 as covering non-statutory subject matter. *See In re Nuijten*, 500 F.3d 1346, 1356-57 (Fed. Cir. 2007) (transitory embodiments are not directed to statutory subject matter) and *Interim Examination Instructions for Evaluating Subject Matter Eligibility Under 35 U.S.C. § 101*, Aug. 24, 2009; p. 2.

The Examiner suggests that the Applicant add the limitation "non-transitory" to the computer readable media as recited in the claim(s) 28, 29-31, 35 in order to properly render the claim(s) in statutory form in view of their broadest reasonable interpretation in light of the originally filed specification.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

Christfort et al (US Pub 2002/0078168)

Uszok et al (US Pub 20040205772)

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 101

A new grounds of rejection under 35 U.S.C. 101 has been added for claims 28-31 and 35.

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claim(s) 28, 29-31, 35 are rejected under 35 USC 101 since the claims are directed to non-statutory subject matter. Claim(s) 29-31 recite computer readable media which appear to cover both transitory and non-transitory embodiments. The United States Patent and Trademark Office (USPTO) is required to give claims their broadest reasonable interpretation consistent with the specification during proceedings before the USPTO. *See In re Zletz*, 893 F.2d 319 (Fed. Cir. 1989) (during patent examination the pending claims must be interpreted as broadly as their terms reasonably allow). The broadest reasonable interpretation of a claim drawn to a computer readable medium (also called machine readable medium and other such variations) typically covers forms of non-transitory tangible media **and** transitory propagating signals *per se* in view of the ordinary and customary meaning of computer readable media, particularly when the specification is silent. *See* MPEP 2111.01. When the broadest reasonable interpretation of a claim covers a signal *per se*, the claim **must** be rejected under 35 U.S.C. § 101 as covering non-statutory subject matter. *See In re Nuijten*, 500 F.3d 1346, 1356-57 (Fed. Cir. 2007) (transitory embodiments are not directed to statutory subject matter) and *Interim Examination Instructions for Evaluating Subject Matter Eligibility Under 35 U.S.C. § 101*, Aug. 24, 2009; p. 2.

The Examiner suggests that the Applicant add the limitation “non-transitory” to the computer readable media as recited in the claim(s) 28, 29-31, 35 in order to properly

render the claim(s) in statutory form in view of their broadest reasonable interpretation in light of the originally filed specification.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of

the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

Claims 1-33, 35 are rejected under 35 U.S.C. 103(a) as being unpatentable by Christfort et al (US Pub 2002/0078168) hereafter known as Christfort, in view of Uszok et al (US Pub 20040205772) hereafter known as Uszok.

Consider Claim 29, Christfort disclosed A computer readable media storing instructions to be executed by a processor of a computer system (Christfort, [0298]), said processor of the computer system executing an integrated development environment (IDE) for generating code for executing in a client-server environment (Christfort, [0080], Christfort discloses an IDE for executing code), to: process an input object identifying code for executing on one of a plurality of servers (Christfort, [0022], Christfort disclosed on identifying several types of input objects which can be used for coding for the application) said processing using a view list of at least one input object element (Christfort, [0022], Christfort discloses on view several input object codes which are presented to the user), each input object element processing a type of code

identified by the input object to output a deployable object (Christfort, [0022], Christfort discloses on how the application code is selected by an object on the interface, and the application code may be executed in a response to a request for a service from an end user); process the deployable object using a server list of at least one server element to determine the one of the plurality of servers for executing the code (Christfort, [0062], Christfort does show the server list for example containing host servers on a portal page), each server element enabling the deployable object to execute on a particular server and outputting a launchable object (Christfort, [0094]-[0095], Christfort disclosed on how objects/created applications are launched via the system); and process the launchable object using a launcher list of at least one client element to determine a client for launching the code on the one of the
plurality of servers (Christford, [0093], Christfort indicates on how portal-to-go XML document or application program containing the code generates the output, and how the output is launched by the system).

But Christford does not explicitly disclose said instructions defining an extensible mechanism for executing said code on a server that, when deployed on said computer system, adapts said IDE for handling new code types

Nonetheless, Uszok discloses said instructions defining an extensible mechanism for executing said code on a server that (Uszok, [0129], Uszok, discloses how the SDK can be used to generate a bot code “executable code” mechanism), when deployed on said computer system, adapts said IDE for handling new code types

(Uszok, [0129], Uszok discloses on how the bot code is able to implement a selection of predetermined protocols set in the SDK).

Both Chirstford, and Uszok provide features related to SDK/IDE management. Therefore one of ordinary skill in the art would have been motivated to combine the teachings since both are within the same environment.

Therefore, it would have been obvious to a person skilled in the art at the time of the invention was made to incorporate executable code mechanism by the SDK/IDE, taught by Uszok, in the system of Christford for the purpose of efficient code execution mechanism.

Claim 1, has similar limitations as Claim 29, therefore it is rejected under the same rational as Claim 1.

Consider Claim 2, Christfort-Uszok disclosed method of claim 1 wherein processing the input object to identify the code for executing on the one of the plurality of servers (Christfort, [0022], Christfort disclosed on identifying several types of input objects which can be used for coding for the application) includes using a view list of at least one input element for processing a type of code identified by the input object (Christfort, [0022], Christfort discloses on view several input object codes which are presented to the user), processing the generated code includes using a

server list of at least one server element for determining the one of the plurality of servers (Christford, [0093], Christfort indicates on how portal-to-go XML document or application program containing the code generates the output, and how the output is launched by the system), and identifying the one of the plurality of client applications includes using a launcher list of at least one client element for launching the one of the plurality of client applications (Christfort, [0062], Christfort does show the server list for example containing host servers on a portal page).

Consider Claim 3, Christfort-Uszok disclosed method of claim 2 wherein at least one of the view list (Christfort, [0022], Christfort discloses on view several input object codes which are presented to the user), server list (Christfort, [0062], Christfort does show the server list for example containing host servers on a portal page) and launcher list is extensible to accommodate additional respective elements (Christford, [0093], Christfort indicates on how portal-to-go XML document or application program containing the code generates the output, and how the output is launched by the system).

Claim 4, has similar limitations as Claim 3, therefore it is rejected under the same rational as Claim 3.

Claim 5, has similar limitations as Claim 3, therefore it is rejected under the same rational as Claim 3.

Consider Claim 6, Christfort-Uszok disclosed the method of Claim 1, wherein processing the input object comprises (Christfort, [0022], Christfort disclosed on identifying several types of input objects which can be used for coding for the application): analyzing the input object to determine an input object element for processing the input object (Christfort, [0080], Christfort discloses on what the input object is); and processing the input object using the determined input object element (Christfort, [0086], Christfort discloses on how the object code is created and developed).

Claim 7, has similar limitations as Claim 6, therefore it is rejected under the same rational as Claim 6.

Consider Claim 8, Christfort-Uszok disclosed the method of Claim 1, wherein the processing the generated code comprises: analyzing a server element for enabling a deployable object (Christfort, [0087]-[0088], Christfort disclosed on how the portal XML to go is analyzed); and processing the deployable object using the determined server element (Christfort, [0093], Christfort disclosed on how the object is deployed with the aid of the XML document).

Consider Claim 9, Christfort-Uszok, Christfort disclosed the method of Claim 8 including processing user input (Christfort, [0091], Christfort discloses on how user input is obtained) to determine the server element (Christfort, [0091], [0093]).

Consider Claim 10, Christfort-Uszok disclosed the method of claim 1 wherein identifying
the one of the plurality of client applications (Christfort, [0095], Christfort disclosed on which identifying the list of applications available) comprises: analyzing a launchable object to determine a client element for processing the launchable object (Christfort, [0094], Christfort disclosed on how the newly created application is launched); and processing the launchable object using the determined client element (Christfort, [0094]-[0095]).

Consider Claim 11, Christfort-Uszok disclosed the method of claim 10, including processing user input to determine the server element (Christford, [0091]).

Claim 12, has similar limitations as Claim 29, therefore it is rejected under the same rational as Claim 29.

Claim 13, has similar limitations as Claim 2, therefore it is rejected under the same rational as Claim 2.

Claim 14, has similar limitations as Claim 3, therefore it is rejected under the same rational as Claim 3.

Consider Claim 15, Christfort-Uszok disclosed the extensible mechanism of Claim 12 wherein said server mechanism comprises a server list of at least one server element (Christford, [0075]-[0076]), each server element enabling the deployable object to execute on a particular server and processing the deployable object for outputting a launchable object (Christford, [0076]).

Claim 16, has similar limitations as Claim 3, therefore it is rejected under the same rational as Claim 3.

Claim 17, has similar limitations as Claim 10, therefore it is rejected under the same rational as Claim 10.

Claim 18, has similar limitations as Claim 3, therefore it is rejected under the same rational as Claim 3.

Consider Claim 19, Christfort-Uszok disclosed the extensible mechanism of claim 12 wherein said extensible mechanism is adapted to launch the one of the plurality of client applications (Christfort, [0095], Christfort disclosed on which identifying the list of applications available) determined in response to the launchable object for executing the code on the one of the plurality of servers (Christford, [0093], Christfort indicates on how portal-to-go XML document or application program containing the code generates the output, and how the output is launched by the system).

Consider Claim 20, Christfort-Uszok disclosed extensible mechanism of claim 12 wherein at least one of said view mechanism, server mechanism, and launcher mechanism (Christfort, [0022], Christfort discloses on view several input object codes which are presented to the user) is extensible whereby said view mechanism is extensible to accommodate a plurality of code types (Christfort, [0022]), said server mechanism is extensible to accommodate a plurality of servers (Christfort, [0062], Christfort does show the server list for example containing host servers on a portal page) and said launcher mechanism is extensible to accommodate a plurality of client applications (Christfort, [0095]-[0096]).

Consider Claim 21, Christfort-Uszok disclosed extensible mechanism of claim 12 wherein said view mechanism (Christfort, [0022], Christfort discloses on view several input object codes which are presented to the user) is adapted to analyze the input

object to determine an input object element for processing the input object and process the input object using the determined input object element (Christford, [0080]-[0084], Christford discloses on how the input entered is analyzed and processed by the system).

Claim 22, has similar limitations as Claim 21, therefore it is rejected under the same rational as Claim 21.

Claim 23, has similar limitations as Claim 15, therefore it is rejected under the same rational as Claim 15.

Claim 24, has similar limitations as Claim 23, therefore it is rejected under the same rational as Claim 23.

Consider Claim 25, Christfort-Uszok disclosed the extensible mechanism of claim 21 wherein said launcher mechanism (Christfort, [0094]-[0095]) is adapted to analyze the launchable object to determine a client element for processing the launchable object (Christfort, [0091]); and process the launchable object using the determined client element (Christford, [0090]-[0091]).

Consider Claim 26, Christfort-Uszok disclosed the extensible mechanism of claim 25 wherein said launcher mechanism is further adapted for processing user input to determine the server element (Christford, [0091]).

Consider Claim 27, Christfort-Uszok disclosed extensible mechanism of claim 12 wherein said extensible mechanism is adapted to be integrated into an integrated development environment (Christfort, [0080]).

Consider Claim 28, Christfort-Uszok disclosed a computer program product embodied in a computer readable medium having instructions that are to be executed by a processor to have a computer system perform a method in accordance with claim 1 (Christfort, [0298]).

Consider Claim 30, Christfort-Uszok disclosed computer readable media (Christfort, [0298]). of claim 29 wherein said IDE (Christfort, [0080]) is further adapted for modifying at least one of the view list, server list and launcher list (Christfort, [0084]).

Claim 31, has similar limitations as Claim 30, therefore it is rejected under the same rational as Claim 30.

Consider Claim 32, Christford disclosed: maintaining at least one of: a view list of at least one input object element (Christford, [0080], Christford discloses a list of inputs which are available to the user), each input object element processing a type of code identified by the input object to output a deployable object (Christford, [0080]-[0084]) a server list of at least one server element to determine one of a plurality of servers for executing the code (Christford, [0091], Christford disclosed on any numbers of servers can be used), each server element enabling the deployable object to execute on a particular server and outputting a launchable object (Christford, [0095], Christfort discloses on how a objects are launched); and a launcher list of at least one client element to determine one of a plurality of client applications (Christford, [0076]) for launching the code on the one of the plurality of servers (Christfort, [0062], Christfort does show the server list for example containing host servers on a portal page).

But Christford does not explicitly disclose said instructions defining an extensible mechanism for executing said code on a server that.

Nonetheless, Uszok discloses said instructions defining an extensible mechanism for executing said code on a server that (Uszok, [0129], Uszok, discloses how the SDK can be used to generate a bot code “executable code” mechanism),

Both Chirstford, and Uszok provide features related to SDK/IDE management. Therefore one of ordinary skill in the art would have been motivated to combine the teachings since both are within the same environment.

Therefore, it would have been obvious to a person skilled in the art at the time of the invention was made to incorporate executable code mechanism by the SDK/IDE, taught by Uszok, in the system of Christford for the purpose of efficient code execution mechanism.

Consider Claim 33, Christford-Uszok disclosed the method of claim 32 wherein the step of maintaining comprises at least one of: generating a respective element; adding a respective element; configuring a respective element; and deleting a respective element from at least one of the view list (Christford, [0080], Christford disclosed on how elements can be entered/modified when being configured to be used in the system), server list (Christford, [0075]-[0076], [0095] Christford disclosed on which server to be used), and launcher list (Christford, [0095], gives the option to launch a specific application).

Consider Claim 35, Chirstford-Uszok discloses the computer readable media of claim 29, further comprising: perform a compatibility test of the input object, deployable

object and launchable object prior to processing the input object; display a result of the compatibility test to the user (Uszok, [0073], Uszok discloses on the compatibility of bot executable code prior to download, to make sure compatibility with the system).

(10) Response to Argument

Independent Claims

For Claim 29 which is representative of independent claims 1, 12, 32, applicant argues that the combination of Chirstford-Uszok does not teach "instructions defining an extensible mechanism for executing said code on a server that, when deployed on the computer system, adapts the integrated development environment for handling new code types.

In response to applicant's arguments, the arguments stated by the appellant for the independent claims 1, 12, 29, and 32 are directed towards the preamble of the claims. The recitation "instructions defining an extensible mechanism for executing said code on a server that, when deployed on the computer system, adapts the integrated development environment for handling new code types" has not been given patentable weight because the recitation occurs in the preamble. A preamble is generally not accorded any patentable weight where it merely recites the purpose of a process or the intended use of a structure, and where the body of the claim does not depend on the

preamble for completeness but, instead, the process steps or structural limitations are able to stand alone. See *In re Hirao*, 535 F.2d 67, 190 USPQ 15 (CCPA 1976) and *Kropa v. Robie*, 187 F.2d 150, 152, 88 USPQ 478, 481 (CCPA 1951). Support can be seen in MPEP 2111.02.

MPEP 2111.02 [R-3] Effect of Preamble

PREAMBLE STATEMENTS RECITING PURPOSE OR INTENDED USE

The claim preamble must be read in the context of the entire claim. The determination of whether preamble recitations are structural limitations or mere statements of purpose or use "can be resolved only on review of the entirety of the [record] to gain an understanding of what the inventors actually invented and intended to encompass by the claim." Corning Glass Works, 868 F.2d at 1257, 9 USPQ2d at 1966. If the body of a claim fully and intrinsically sets forth all of the limitations of the claimed invention, and the preamble merely states, for example, the purpose or intended use of the invention, rather than any distinct definition of any of the claimed invention's limitations, then the preamble is not considered a limitation and is of no significance to claim construction. Pitney Bowes, Inc. v. Hewlett-Packard Co., 182 F.3d 1298, 1305, 51 USPQ2d 1161, 1165 (Fed. Cir. 1999). See also Rowe v. Dror, 112 F.3d 473, 478, 42 USPQ2d 1550, 1553 (Fed. Cir. 1997) ("where a patentee defines a structurally complete invention in the claim body and uses the preamble only to state a purpose or intended use for the invention, the preamble is not a claim limitation"); Kropa v. Robie, 187 F.2d at 152, 88 USPQ2d at 480-81 (preamble is not a limitation where claim is directed to a product and the preamble merely recites a property inherent in an old product defined by the remainder of the claim); STX LLC. v. Brine, 211 F.3d 588, 591, 54 USPQ2d 1347, 1350 (Fed. Cir. 2000) (holding that the preamble phrase "which provides improved playing and handling characteristics" in a claim drawn to a head for a lacrosse stick was not a claim limitation). Compare Jansen v. Rexall Sundown, Inc., 342 F.3d 1329, 1333-34, 68 USPQ2d 1154, 1158 (Fed. Cir. 2003) (In a claim directed to a method of treating or preventing pernicious anemia in humans by administering a certain vitamin preparation to "a human in need thereof," the court held that the preamble is not merely a statement of effect that may or may not be desired or appreciated, but rather is a statement of the intentional purpose for which the method must be performed. Thus the claim is properly interpreted to mean that the vitamin preparation must be administered to a human with a recognized need to treat or prevent pernicious anemia.); In re Cruciferous Sprout Litig., 301 F.3d 1343, 1346-48, 64 USPQ2d 1202, 1204-05 (Fed. Cir. 2002) (A claim at issue was directed to a method of preparing a food rich in glucosinolates wherein cruciferous sprouts are harvested prior to the 2-leaf stage. The court held that the preamble phrase "rich in glucosinolates" helps define the claimed invention, as evidenced by the specification and prosecution history, and thus is a limitation of the claim (although the claim was anticipated by prior art that produced sprouts inherently "rich in glucosinolates")).

During examination, statements in the preamble reciting the purpose or intended use of the claimed invention must be evaluated to determine whether the recited purpose or intended use results in a structural difference (or, in the case of process claims, manipulative difference) between the claimed invention and the prior art. If so, the

recitation serves to limit the claim. See, e.g., In re Otto, 312 F.2d 937, 938, 136 USPQ 458, 459 (CCPA 1963) (The claims were directed to a core member for hair curlers and a process of making a core member for hair curlers. Court held that the intended use of hair curling was of no significance to the structure and process of making.); In re Sinex, 309 F.2d 488, 492, 135 USPQ 302, 305 (CCPA 1962) (statement of intended use in an apparatus claim did not distinguish over the prior art apparatus). If a prior art structure is capable of performing the intended use as recited in the preamble, then it meets the claim. See, e.g., In re Schreiber, 128 F.3d 1473, 1477, 44 USPQ2d 1429, 1431 (Fed. Cir. 1997) (anticipation rejection affirmed based on Board's factual finding that the reference dispenser (a spout disclosed as useful for purposes such as dispensing oil from an oil can) would be capable of dispensing popcorn in the manner set forth in appellant's claim 1 (a dispensing top for dispensing popcorn in a specified manner)) and cases cited therein. See also MPEP § 2112 - § 2112.02.
>However, a "preamble may provide context for claim construction, particularly, where … that preamble's statement of intended use forms the basis for distinguishing the prior art in the patent's prosecution history." Metabolite Labs., Inc. v. Corp. of Am. Holdings, 370 F.3d 1354, 1358-62, 71 USPQ2d 1081, 1084-87 (Fed. Cir. 2004). The patent claim at issue was directed to a two-step method for detecting a deficiency of vitamin B12 or folic acid, involving (i) assaying a body fluid for an "elevated level" of homocysteine, and (ii) "correlating" an "elevated" level with a vitamin deficiency. 370 F.3d at 1358-59, 71 USPQ2d at 1084. The court stated that the disputed claim term "correlating" can include comparing with either an unelevated level or elevated level, as opposed to only an elevated level because adding the "correlating" step in the claim during prosecution to overcome prior art tied the preamble directly to the "correlating" step. 370 F.3d at 1362, 71 USPQ2d at 1087. The recitation of the intended use of "detecting" a vitamin deficiency in the preamble rendered the claimed invention a method for "detecting," and, thus, was not limited to detecting "elevated" levels. Id .
See also Catalina Mktg. Int 'l v. Coolsavings.com, Inc., 289 F.3d at 808-09, 62 USPQ2d at 1785 ("[C]lear reliance on the preamble during prosecution to distinguish the claimed invention from the prior art transforms the preamble into a claim limitation because such reliance indicates use of the preamble to define, in part, the claimed invention.…Without such reliance, however, a preamble generally is not limiting when the claim body describes a structurally complete invention such that deletion of the preamble phrase does not affect the structure or steps of the claimed invention." Consequently, "preamble language merely extolling benefits or features of the claimed invention does not limit the claim scope without clear reliance on those benefits or features as patentably significant."). In Poly-America LP v. GSE Lining Tech. Inc., 383 F.3d 1303, 1310, 72 USPQ2d 1685, 1689 (Fed. Cir. 2004), the court stated that "a [r]eview of the entirety of the '047 patent reveals that the preamble language relating to blown-film' does not state a purpose or an intended use of the invention, but rather discloses a fundamental characteristic of the claimed invention that is properly construed as a limitation of the claim.…'" Compare Intirtool, Ltd. v. Texar Corp., 369 F.3d 1289, 1294-96, 70 USPQ2d 1780, 1783-84 (Fed. Cir. 2004) (holding that the preamble of a patent claim directed to a "hand-held punch pliers for simultaneously punching and connecting overlapping sheet metal" was not a limitation of the claim because (i) the body of the claim described a "structurally complete invention" without the preamble, and (ii) statements in prosecution history referring to "punching and connecting" function of invention did not constitute "clear reliance" on the preamble needed to make the preamble a limitation)

Applicant argues that the combination of Chirstford-Uszok does not teach the feature of processing an input object identifying code for executing on one of a plurality of servers, said processing using a view list of at least one input object element, each input object element processing a type of code identified by the input object to output a deployable object.

Examiner states that the combination of Chirstford-Uszok does indeed teach the limitation “input object identifying code for executing on one of a plurality of servers, said processing using a view list of at least one input object element, each input object element processing a type of code identified by the input object to output a deployable object”. Chirstford in [0022] teaches on how request is received from the user's browser of an application development interface. The interface can handle several types of objects, including an edit field for typing code for an application. Christford discloses on how code is edited and be used to create an application within the IDE stated by Chirstford. Since the code can be displayed by the IDE, therefore it can be deployed by the system when it compiled or executed in the system. Further support can be seen in Chirstford [0059]—where online SDK can be used by developers to create, test, modify and deploy applications. And especially deploying applications where the client does not any client side software to run it - hence executing the application in the server at the same time providing access on the client side of the system. For example deployable object can be the program created in the system by the SDK to be executed in the server side for the client to have access to it. Applicant argues that Christford does not disclose a view list of input object element in the view list outputs a deployable

object. In [0070]-[0072] of Chirstford, it is clearly seen that elements of the objects can be seen and be deployed to the client.

Applicant argues that the combination of Christfort-Uszok further does not teach the feature of processing the deployable object using a server list of at least one server element to determine the one of the plurality of servers for executing the code. The combination of Christford-Uszok does indeed teach deployable object using a server list of at least one server element to determine the one of the plurality of servers for executing the code. Christford discloses on how the applications can be created and deployed throughout the network (Chirstford, [0062], [0090]-[0091]). The client can be a server, as it is requesting a service by the server which is hosting the service (Chirstford, [0076], Chirstford discloses on how – for example “the map service provider may link to another hosted or shared hosted application provided by a weather information service provided that is associated with the host server”. Further support can be seen when the application is deployed as a shared hosted application (Christford, [0093]-[0094], Chirstford discloses on deployable object "the application/shared hosted application" for example can be executed on other servers.

Applicant argues that the combination of Chirstford-Uszok does not teach processing the launchable object using a launcher list of at least one client element to determine a client for launching the code on the plurality of servers. Christford in [0093]-[0095] discloses on how the client is able to choose the service which contains the launchable object and be able to execute it. In Chirstford [0095], Chirstford

discloses on how a user is presented with a list of available services available from the server (launcher list), and may select a service which can be executed.

Dependent Claims

For claim 2, Applicant argues that the combination of Christford-Uszok does not teach processing the input object to identify the code for executing on the one of the plurality of servers includes using a view list of at least one input element for processing a type of code identified by the input object, processing the generated code includes using a server list of at least one server element for determining the one of the plurality of servers, and identifying the one of the plurality of client applications includes using a launcher list of at least one client element for launching the one of the plurality of client applications.

The combination of Chirstford-Uszok does indeed teach the following limitation as stated in Claim 2. The combination of Christford-Uszok does indeed teach deployable object using a server list of at least one server element to determine the one of the plurality of servers for executing the code. Christford discloses on how the applications can be created and deployed throughout the network (Chirstford, [0062], [0090]-[0091]). The client can be a server, as it is requesting a service by the server which is hosting the service (Chirstford, [0076], Chirstford discloses on how – for example “the map service provider may link to another hosted or shared hosted application provided by a weather information service provided that is associated with

the host server". Further support can be seen when the application is deployed as a shared hosted application (Christford, [0093]-[0094], Chirstford discloses on deployable object "the application/shared hosted application" for example can be executed on other servers. Christford in [0093]-[0095] discloses on how the client is able to choose the service which contains the launchable object and be able to execute it. In Chirstford [0095], Chirstford discloses on how a user is presented with a list of available services available from the server (launcher list), and may select a service which can be executed. And using a server list of at least one server element for determining the one of the plurality of servers, the combination of Christford-Uszok does indeed teach deployable object using a server list of at least one server element to determine the one of the plurality of servers for executing the code. Christford discloses on how the applications can be created and deployed throughout the network (Chirstford, [0062], [0090]-[0091]). The client can be a server, as it is requesting a service by the server which is hosting the service (Chirstford, [0076], Chirstford discloses on how – for example "the map service provider may link to another hosted or shared hosted application provided by a weather information service provided that is associated with the host server", and identifying the one of the plurality of client applications includes using a launcher list of at least one client element for launching the one of the plurality of client applications - Christford in [0093]-[0095] discloses on how the client is able to choose the service which contains the launchable object and be able to execute it. In Chirstford [0095], Chirstford discloses on how a user is presented with a list of available

services available from the server (launcher list), and may select a service which can be executed.

Applicant argues for claim 13 that combination of Chirstford-Uszok does not teach a view list of at least one input object element, each input object element processing a type of code identified by the input object for outputting the deployable object. The combination of Chirstford-Uszok does indeed teach view list of at least one input object element and each input object element processing a type of code identified by the input object for outputting the deployable object (Chirstford, [0070], Christford discloses on how output for a transforming application output that is inputed as a input object into output which is tailored or customized based on parameters or conditions associated with the service request). Chirstford discloses on how output of an application is modified with code and used as input in the system to further create the desired deployable object.

For Claims 3, 14, 16, 18, the applicant argues that Christford-Uszok does not teach view list, server list, and launcher list. The combination of Chirstford-Uszok does indeed teach these types of list - The combination of Christford-Uszok does indeed teach deployable object using a server list of at least one server element to determine the one of the plurality of servers for executing the code. Christford discloses on how the applications can be created and deployed throughout the network (Chirstford,

[0062], [0090]-[0091]). The client can be a server, as it is requesting a service by the server which is hosting the service (Chirstford, [0076], Chirstford discloses on how – for example "the map service provider may link to another hosted or shared hosted application provided by a weather information service provided that is associated with the host server". Further support can be seen when the application is deployed as a shared hosted application (Christford, [0093]-[0094], Chirstford discloses on deployable object "the application/shared hosted application" for example can be executed on other servers. Christford in [0093]-[0095] discloses on how the client is able to choose the service which contains the launchable object and be able to execute it. In Chirstford [0095], Chirstford discloses on how a user is presented with a list of available services available from the server (launcher list), and may select a service which can be executed.

For Claim 6 and 7, the applicant argues that the combination of Chirstford-Uszok does not explicitly teach processing of an input object for the purpose of subsequently identifying code to be executed by a server. The combination of Chirstford-Uszok does indeed teach processing of an input object element processing a type of code identified by the input object for outputting the deployable object (Chirstford, [0070], Christford discloses on how output for a transforming application output that is inputted as a input object into output which is tailored or customized based on parameters or conditions associated with the service request). Chirstford discloses on how output of an application is modified with code and used as input in the system to further create the desired deployable object.

For Claims 8, 9, and 15, the applicant argues that the combination of Chirstford-Uszok does not teach the analysis of server side elements or teach wherein such analysis is used for enabling a deployable object. In Chirstford, [0091]-[0092], Chirstford discloses on how analysis is determined for enabling a deploying object, if the deployable object is not tested properly, the object will fail to deploy. Furthermore, one of the embodiments in Chirstford – the invention allows the input object element being processed into a type of code identified by the input object for outputting the deployable object (Chirstford, [0070], Christford discloses on how output for a transforming application output that is inputted as a input object into output which is tailored or customized based on parameters or conditions associated with the service request).

For Claims 10, 11 and 17, the applicant argues that the combination of Christford-Uszok does not teach any analysis of a launchable object to determine a client application for processing the launchable object. In Chirstford, [0091]-[0092], Chirstford discloses on how analysis is determined for enabling a deploying object, if the deployable object is not tested properly, the object will fail to deploy. Once the object is deployed successfully, it is inherent that it will be able to launch properly.

For claim 35, the applicant argues that combination of Chirstford-Uszok does not teach the compatibility test of objects prior to processing the input object. In Uszok

[0073], the compatibility of the bot code is downloaded and verified. The bot code is treated as object code and it is tested for compatibility prior to processing the bot code.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

This examiner's answer contains a new ground of rejection set forth in section **(9)** above. Accordingly, appellant must within **TWO MONTHS** from the date of this answer exercise one of the following two options to avoid *sua sponte* **dismissal of the appeal** as to the claims subject to the new ground of rejection:

(1) **Reopen prosecution.** Request that prosecution be reopened before the primary examiner by filing a reply under 37 CFR 1.111 with or without amendment, affidavit or other evidence. Any amendment, affidavit or other evidence must be relevant to the new grounds of rejection. A request that complies with 37 CFR 41.39(b)(1) will be entered and considered. Any request that prosecution be reopened will be treated as a request to withdraw the appeal.

(2) **Maintain appeal.** Request that the appeal be maintained by filing a reply brief as set forth in 37 CFR 41.41. Such a reply brief must address each new ground of rejection as set forth in 37 CFR 41.37(c)(1)(vii) and should be in compliance with the other requirements of 37 CFR 41.37(c). If a reply brief filed pursuant to 37 CFR

41.39(b)(2) is accompanied by any amendment, affidavit or other evidence, it shall be treated as a request that prosecution be reopened before the primary examiner under 37 CFR 41.39(b)(1).

Extensions of time under 37 CFR 1.136(a) are not applicable to the TWO MONTH time period set forth above. See 37 CFR 1.136(b) for extensions of time to reply for patent applications and 37 CFR 1.550(c) for extensions of time to reply for ex parte reexamination proceedings.

Respectfully submitted,

Anish Sikri

A Technology Center Director or designee must personally approve the new ground(s) of rejection set forth in section (9) above by signing below.

Conferees:

/George C Neurauter, Jr./

Primary Examiner, Art Unit 2443

/Tonia LM Dollinger/

Supervisory Patent Examiner, Art Unit 2443

/Jack Harvey/

Director, Technology Center 2400